



Durham Research Online

Deposited in DRO:

05 August 2010

Version of attached file:

Published Version

Peer-review status of attached file:

Peer-reviewed

Citation for published item:

Heslin, P. J. (2006) 'Review of the Thesaurus Linguae Latinae, 3rd electronic edition.', Bryn Mawr classical review. (2006.02.19).

Further information on publisher's website:

<http://ccat.sas.upenn.edu/bmcr/2006/2006-02-19.html>

Publisher's copyright statement:

Additional information:

Use policy

The full-text may be used and/or reproduced, and given to third parties in any format or medium, without prior permission or charge, for personal research or study, educational, or not-for-profit purposes provided that:

- a full bibliographic reference is made to the original source
- a [link](#) is made to the metadata record in DRO
- the full-text is not changed in any way

The full-text must not be sold in any format or medium without the formal permission of the copyright holders.

Please consult the [full DRO policy](#) for further details.

Bryn Mawr Classical Review

Bryn Mawr Classical Review 2006.02.19

Thesaurus Linguae Latinae, *Third electronic edition*. Munich: K. G. Saur, 2005. ISBN 3-598-40772-6 (CD-ROM). €1,050.00. ISBN 3-598-40841-2 (DVD).

Reviewed by Peter Heslin, University of Durham (p.j.heslin@dur.ac.uk)

Word count: 9102 words

In stages over the last few years, the publishing house of K. G. Saur has been expanding the coverage of its electronic version of the Thesaurus Linguae Latinae (TLL), a venerable printed resource that many people would consider the most important tool we have for the study of the Latin language. With version 3, this electronic edition has now reached a point where it encompasses nearly all of the currently realized portion of the printed text, so this is an opportune moment for a review. Here is a summary of the positive findings: most of the massive task of digitization has been done very accurately; the technical foundations of the project are extremely sound; and in many ways the electronic version is already more pleasant and useful to consult than the printed volumes. For these reasons, those Latinists who can afford it will want to buy it. Classics departments ought to consider this an essential resource, to which all staff and postgraduate students must have access. On the other hand, there are negatives, too: at present the electronic version is available for Microsoft Windows only; the user interface is slow and cumbersome; a handful of entries are missing; and the networked version is tedious to install on multiple computers. Fortunately, the excellent technical foundations of the project mean that these deficiencies can easily be remedied, and if Saur wishes to do so, it can very quickly bring out a cross-platform, web-enabled version of the TLL, which would make this important tool much more widely and easily available to students of Latin at all levels.

It might be a good idea to begin by giving a general idea of what the TLL is and what it is not. On more than one occasion, I have heard professional classicists using the term "TLL" erroneously to indicate the CD of Latin texts published by the Packard Humanities Institute. This confusion arises from the assumption that the TLL is symmetrical to the TLG, or Thesaurus Linguae Graecae, which is a collection of texts on a CD (or has been up to version E -- future upgrades to the TLG are apparently going to be limited to web subscribers). Now that the TLL is itself available in an electronic version, this confusion between the nature of the two projects is likely to arise even more often. By contrast with the TLG, the TLL is not simply a collection of texts, but is a lexicon, whose primary existence has been as a massive printed reference work. Since the year 1900, it has been published progressively in fascicles, and the work, which has reached the letter "p" (having skipped "n"), continues to this day; when "p" is finished, "n" and "r" will be done simultaneously. The third electronic edition includes all of the printed text, except for the most recently published "p" fascicles (the electronic edition goes up to "piulcus" in the first part of the "p" volume, and up to "propos" in the second). The few volumes of the Onomasticon that have been published in the printed version are also available in the electronic version.

When you buy the electronic TLL, you get an application that is straightforward to install on a

single PC, and which a user of moderate technical ability should be able to figure out how to use, without needing special instruction. It is therefore very suitable for installation in a departmental library or computer room. The initial screen is confusing (see below), but it should be possible for most users to figure out how to perform basic tasks by trial and error. The help facility is exceptionally thorough and clear. The prospective buyer should be aware that Saur is expecting to publish the fourth version of the electronic TLL around May of this year, and so in order to avoid the inconvenience of upgrading, it might be worth waiting for that release. The next version will include the few entries inadvertently left out of the current version (see below), the "Praemonenda", and will update the letter "p" by including fascicle XV of vol. X, part 1, and fascicle XIV of vol. X, part 2.

The lexicographic methodology of the TLL has been described in many other places and will not be repeated in detail here.¹ In brief, the process of compiling each entry begins with a comprehensive collation of all of the instances of a given word in all of the texts covered by the lexicon, which include all literature down to Apuleius, and significant texts thereafter to Isidore of Seville, and all inscriptions down to the early Principate, and excerpts thereafter. These instances are then categorized systematically, and an article consists mainly of the presentation of all of the citations in their semantic groupings, or in the case of more commonly used words, a selection of the most interesting citations. The result is that the TLL is a set of data already arranged for human perusal, and, unlike the TLG CD, does not require complex search software to be useful. Its user interface, so to speak, consists of the alphabetical ordering of entries, the cross-references, and the line and column numbers that pinpoint each citation. This interface has been worn smooth by the hands of generations of scholars, and the most important task for the user interface of an electronic edition should be to make this traditional usage easy. This may seem a painfully obvious observation, but it seems to me that the developers of the user interface for the electronic edition have overlooked the importance of this fundamental fact, and, as we shall see, its deficiencies can plausibly be attributed to the misconception that the foremost requirements of an electronic TLL are complex and powerful search facilities along the lines of what the TLG requires.

Pricing, Ordering and Installation

I am reviewing the copy of the TLL that I ordered for my own department; I did not ask for a reviewer's copy. The retail prices listed by Saur are:²

EUR 1,050.00 for customers without a discount;

EUR 840.00 for those who already subscribe to the print edition;

EUR 680.00 for those upgrading from an earlier electronic edition;

EUR 550.00 for those upgrading who also subscribe to the print edition.

The same price applies, regardless of whether you order it on one DVD or three CDs. Since the Bibliotheca Teubneriana Latina CD, which is also published by Saur and is listed on the same page as the one from which I ordered the TLL,³ requires that the CD always be inserted in the computer in order to use the software, I assumed and feared that the same would be true of the TLL. I therefore ordered the DVD version rather than the multiple CD version, since I did not want to be swapping CDs around while accessing the TLL. So I was annoyed at first to discover that I was sent the multiple CD version by mistake, but immediately relieved to discover that swapping around the CDs is only necessary on installation; all of the data can be installed on the hard disk, and accessed from there. This is an important point, since any

requirement to access a database from a CD or DVD would render it much less useful, especially in an open computer room used by students; experience with the Teubner disk bears this out.

The minimum system requirements are allegedly Windows 98 or better on a Pentium I with 128 MB of memory (256 MB or more is recommended) and a little under 3 GB of disk space, when the TLL data and the search software and its prerequisites are added together. As I do not use Windows myself, I installed the software on a machine in our departmental computer room, a 733 Mhz Pentium III with 512 MB of memory, running Windows 2000. This is a five-year-old computer, and so it is not surprising that the TLL runs quite slowly on it (see below for specifics), even though it handsomely exceeds the stated minimum specification; I suspect that it would be excruciatingly slow on an older machine.

Installation was straight-forward, and the instructions are clear; the process requires administrator rights. English and German versions of the user interface are provided. In addition to the search software, version 6 of the Internet Explorer browser is installed if it is not present already. A special-purpose font (called "TLL") is also installed. During installation, you have the option of installing the Thesaurus data themselves (this is on disk 3 if you have the CD version). Most users will want to make sure to do this, to take advantage of the greater speed and convenience of accessing the data from the hard disk, rather than the CD.

In addition to the single-user prices quoted above, Saur advertises network prices: a surcharge of 50% for 2-4 simultaneous users, 100% for 5-10 users, 150% for 11-25 users, and 200% for more than 26 users. I did not know about this when I ordered the TLL, so I have not tried a network installation, but the instructions that come with the disk(s) explain how it is supposed to work. Essentially, you go through the same process as for a single-user install in the case of each client machine, except that the database itself can be shared among the clients. In these days of large, cheap hard disks, that is not really much of an advantage. On the other hand, having to install specialized software for each client machine is painful, particularly on a large, heterogeneous network, such as found in most universities. It means that, if the operating system needs to be reinstalled on any of these computers, the TLL software probably needs to be reinstalled as well. This is the sort of time-consuming scenario that a client-server model is supposed to avoid. Fortunately, a different approach is possible; see below.

It should be noted that it might be possible to run the TLL on a non-Windows machine by using virtualization software, which allows the user to install a copy of Windows that runs within a non-Windows operating system. Saur claims that it has heard reports of success in getting the TLL to run on a Mac by means of this technique, but this option is not for the faint of heart.

The Infrastructure

Rather than beginning at the top, with a discussion of the interface that the user sees when the program starts up, we will proceed from the bottom up, and start with a look at the way the digitization project was carried out, and how the data have been stored. The individual entries in the TLL have each been encoded as Extensible Markup Language (XML) files, in a scheme that is highly reminiscent of the guidelines of the Text Encoding Initiative (TEI) for print dictionaries. At one point the TEI was officially involved with an abortive attempt to digitize the TLL, which seems to have come to nothing, but perhaps it is due to that influence that the XML used in the Saur project has its present shape.⁴ The combination of TEI-flavored XML and Unicode is an excellent choice for many reasons: they are widely-used, open standards which are likely to endure for a long time and thus to ensure that the data remain intelligible

and usable through future changes of technology.

In order to present the data to the viewer, the TLL comes with three examples of Extensible Stylesheet Transformations (XSLT) that can transform the raw XML data into three different types of web (HTML) page for viewing in a browser. One of these stylesheets (ttsarticle.xml) provides an "article view" which attempts to show the data in a format as similar as possible to the printed page. This is essential for checking references to the precise column and line of a given entry, which is the standard mode of citation for the TLL. There is a second stylesheet (ttsoutline.xml), which takes the same raw XML data, and gives an "outline view" of the article. This is where the advantage of having the TLL in an electronic form is most evident, in my view. The structure of a typical TLL entry (though less consistently so in the earlier volumes) is deeply nested. Computers are very good at dealing with such data structures, and indeed XML itself is an example of such a structure, but humans can find them difficult to work with.

In other words, the TLL divides citations into large semantic groups, then each group is divided into sub-groups, which may be divided into sub-sub-groups and so on, often down to more than a dozen or so sub-levels; this is in contrast to a work like the Oxford Latin Dictionary, which gives a long, mostly flat list of definitions. One of the difficulties in consulting the printed TLL is keeping mental track of all of these levels. They are distinguished by varying the style of enumeration, but the lack of indentation can make it hard to see which level is subordinate to which. On the computer screen, where limitations of page length do not apply, the outline mode of the electronic TLL affords a much more synoptic view of each entry.

Anyone who has consulted the Perseus Project on-line version of the Lewis and Short Latin lexicon or the Liddell-Scott-Jones Greek lexicon⁵ will be able to appreciate the advantage of reading an entry in which progressive levels of indentation and ample vertical space allow the structure to become more manifest than can be afforded on the printed page. In addition to showing the internal structure of each entry via indentation, paragraphing and labeling, the outline view includes little boxes next to each sub-heading, which can be used to hide or display the contents of that section, thus folding away from view those parts of the entry that are not of interest. This provides a much clearer picture of the structure of each entry than the printed edition can do, and is a major boon.

The third stylesheet is used to present a "citation view" of the raw XML data. This is a list, alphabetical by author, of all of the citations listed in a given entry. This will be extremely valuable in cases where you want to find out whether the TLL categorizes the particular bit of text you are interested in and, if so, where. At the moment, you have to either skim through the whole entry or guess where the citation might be. The former is not practical with very long entries, and the latter is a problem when the very reason you are consulting the TLL is that you do not understand the usage of a word in a particular context. Because it attempts to categorize each and every interesting usage of every word in the texts it covers, the TLL is a commentary of last resort. If you want a second opinion on the usage in a particular instance, and the commentaries do not exist or let you down, the TLL entry, if it has been written for your word, very likely has something to say.

The TLL is not an absolute authority -- if you look hard enough at the categorization of just about any of the citations, you can develop an argument that it really belongs in two or more categories. The metaphorical play with semantic boundaries that is essential to poetic discourse can be at odds with the lexicographic methodology of the TLL, if rigidly applied; and increasingly its editors have acknowledged and accommodated this sort of ambiguity. Nevertheless, there is a very good chance that, when you are facing a bit of interesting or obscure usage, the TLL will have registered the opinion of some intelligent person on what the

word means in this context. You may not agree, but having another opinion is invaluable, whether it is to suggest the correct answer, or simply to provide a way of clarifying one's own views by contrast.

The problem with treating the TLL thus not as a dictionary but as a vast but specialized commentary on the whole of classical Latin literature lies in the difficulty, except in short entries, in locating a given citation within the text. This problem has now been removed by the electronic TLL. In fact, even without the citation view, this would be possible by opening the entry in article view and then using the search functionality of your browser to look for the ancient author and work in question. The citation view, however, is a much cleaner and more straight-forward way to get this information, and it also provides a handy list of all of the other citations of the word from a given author and work included in the present entry. Citation view also gives a convenient overview of the distribution of usage across works and authors, but an additional stylesheet to present these in chronological as well as alphabetical order would be welcome.

To quibble, there are a few small ways in which the encoding and presentation of the data might be improved. One easy change would be to add yet another stylesheet, which would provide a printable view, just like the article view, but with more printer-friendly settings. Printing the article view from Internet Explorer requires ensuring that one prints only the frame in which the article appears (e.g. by right-clicking on it), and even then the text is uncomfortably large and cuts off the ends of longer lines, at least on the machine and printer I was using. The names of entry compilers, which are right-justified at the end of the article, are in this way invariably cut off. This is an easy change, but an important one, as most people who need to consult an entry intensively would probably prefer to do so on paper. The pages of the printed TLL need to be reduced in size to photocopy onto standard sized paper, and since the fascicles are usually bound into massive volumes, photocopied text near the spine is often illegible. Library copies of the printed TLL often feature cracked spines, torn pages, and broken covers, much of which is probably due to photocopying, a practice which could be superseded by printing articles from the electronic edition.

While we are quibbling, it is worth noting that the <location> element within each XML <xr> element, which gives the precise spot from which a citation comes within a given author and work, is not subdivided hierarchically (it is just text, or mixed content in XML parlance). Thus, in the citation "VERG. Aen. 3, 649", the author and work are marked-up individually, but the book number and line number are lumped together as "3, 649". This would have been a difficult problem to solve, as can be seen if we look at the more complicated citation "SERV. Aen. ad l. 3.649". Here, the TLL gives "SERV." as the author, the work is null, and the location is "Aen. ad l. 3.649", where the "ad l." contains markup that indicates italics; so here we are dealing with something much more complicated than a series of comma-separated numbers. Figuring out how to standardize systems of reference for ancient texts is a big problem, and it is easy to see why Saur did not feel that this was its problem to solve; thus it just took the easy way out. The shame of it is that a system of marking up precise citation information would have provided the opportunity of hyperlinking the electronic TLL to a database of Latin texts, such as the Teubner database, which is also published by Saur. At the moment, clicking on a TLL citation brings you to the entry for that work in the TLL Index Librorum, so that you know from what edition of the text the citation was taken. This is useful, but still more useful would be a link to the passage in the original text.⁶

Another example of a smaller problem arising from the lack of markup within the location information is that the sorting order in citation view is alphabetical, which is fine for the names of authors and works, but it is not right for the mainly numerical locations within each work. A

variety of crude XSLT fixes for this are possible, but these will only work in some cases, and a proper solution would require a hierarchical markup scheme for all of the TLL citations (see appendix below).

To sum up, the electronic TLL provides new ways of using the tool that either were not possible or were very cumbersome with the printed text. The three "views" of each entry are valuable because they work in harmony with the way each entry itself has been structured by its compiler, exposing his or her intent more clearly than was possible on a crowded printed page that could only offer one view of the data. Furthermore, Saur is to be commended for having implemented this functionality on top of an excellent technical infrastructure, which could hardly be bettered. By using open, platform-independent, technologies such as Unicode, TEI-based XML, XSLT and HTML, each in their appropriate role, Saur has ensured that the results of this essential task of digitization will endure.

Proofreading

Of course, the excellence of the technical infrastructure of the electronic TLL is only meaningful if the massive job of digitizing the text of the printed TLL and marking it up was done well. It seems fair to say that the accuracy of the electronic version is excellent, especially given the intricacy and scope of the project. Naturally, there are some errors, but I only discovered one small patch where the accuracy was seriously amiss.

I proofread at random one entry from each of the ten volumes, chosen from entries at least one column long, and generally fitting on one printed page for ease of photocopying. Here are the articles I proofread and the errors of transcription I found:

* alea: no errors

* baca: no errors

* chelys:

col. 1005, l. 74, "voc." should be letter-spaced.

* diligentia:

col. 1174, l. 1, read "cum" for "cam".

* efflagito:

col. 188, l. 14, read "parendi" for "parondi";
col. 188, l. 43, read "studiose" for "etudiose".

* futuo:

col. 1663, l. 37, "dicere" should be italic;
col. 1664, l. 12, read "Crescens" for "rescens";
col. 1664, l. 21, read "hic fu(tu) -i/t" for "hio fu(tu)i/t";
col. 1664, l. 29, remove spaces around the dot in "copo.nam";
col. 1664, l. 34, read "XVI" for "XIV".

* liberatio: no errors

* manubiae:

col. 335, l. 67, read "ornamenta" for "ornameata";
col. 335, l. 72, read "-bIs" for "-bis";
col. 336, ll. 25f, "de" should be letter-spaced (twice).

* ominor: no errors in normal text, but see below.

* pantomimus: no errors.

It would be wrong to draw any sweeping conclusions from such a minute sample, but it is perhaps not surprising that some articles have been proofread slightly better than others. Overall, the standard of transcription is extremely high, and the errors are minor.

There was one surprise. I deliberately included "ominor" in my survey, since that article had fairly lengthy addenda, on account of a section having been left out of the original printing of the fascicle in which it appeared. In Luehken's 2003 review of the first edition of the electronic TLL, he noted problems here (p. 1119). In this edition, the addenda and corrigenda are handled well. In the web page, there is a blue dot (Unicode U+25CF) before the headword which indicates that the article has addenda or corrigenda, and then at the relevant point in the article, there is an icon of a hand with a finger pointing upwards (U+261D), which invites the user to click on it as a link to the extra material.

I looked at this text simply to make sure that the electronic TLL included such addenda; I did not expect it to be any more or less well proofread than any other part. What I found was that the original text of the "ominor" entry was perfect, but the text of the 18 lines of addenda to that entry was proofread to a much lower standard than anything else I looked at:

col 1216:

l. 59, read "humanitate" for "hutnanitate" and "excusaturum" for "excugaturum"; l. 62, there should be a full stop rather than a comma after "subj", and there should be a space after "14,"; l. 67, there should be a space before "1973", there should be no hyphen before "54", and there should be a space after "35,"; l. 68, there should be a full stop rather than a comma after "venisse", and likewise after "patient" in l. 70 and after "cf" in l. 73; also in l. 73, there should be spaces after the commas in "3,61,5", and likewise after "580," and "22," in the next line; l. 75, read "prospera" for "prosper".

It may be that this text slipped through the net because it was an addendum, but it is also possible that other patches of poor proofing may exist in other parts of the TLL.

A few conclusions can be drawn from the observation that the standard of proofreading is excellent but not perfect. The first is that the electronic TLL cannot substitute entirely for the canonical printed version, and so libraries who subscribe to the former should not give up their subscriptions to the latter. Another is that there should be some easy way for readers to report errors of transcription to the publisher. One of the advantages of an electronic edition is that such errors can very easily be fixed, if there is a will to do so. In his review, Luehken pointed out a few typos, and these still have not been corrected in the third edition.⁷ It might therefore be worthwhile if Saur were to provide an e-mail address to which readers of the electronic TLL could direct reports like this.

Missing Entries

Knowing that I was in the process of writing this review, Professor Harry Hine contacted me to report that he had found the entry for the word "circa" missing in his copy of the electronic TLL. He then contacted Saur, who checked, and acknowledged that it had been omitted mistakenly from version 3; they explained that this and 11 other words had been kept to one side for technical reasons, and had been inadvertently omitted from publication. These words are: caph, capillus, caro, caussor, cito, circa, circiter, colonus, commemoro, compages, determinatio, detundo. In an exemplary reaction, Saur has undertaken to provide HTML files of each of the missing entries in their three viewing modes (article, outline and citation) to anyone who has bought the third edition of the electronic TLL and asks for them; these missing entries will be included in the next electronic edition.

In response to a query about this problem, Saur said that it had done a thorough inventory of all the lemmata, and that these twelve were the only ones that were missing. Just to double-check, I put together a long list of Latin words, mostly derived from the Perseus version of the Lewis and Short Latin lexicon, removed the proper names and words beginning with "n" and letters from "p" to "z", and looked to see if any of these words were missing from the electronic TLL. The only ones I found to be missing were already on Saur's list of twelve, so this offers some support to the claim that there are no other entries missing, though it should be said that my checklist was derived from a much smaller collection of lemmata than in the TLL.

HTML Display

As mentioned above, the TLL entries are presented in the form of three different HTML views, and it must be said that the layout of these web pages is not as pleasing to the eye as it could be, but these problems can be easily fixed. In subjective order of severity these are: the use of an oblique roman typeface instead of true italic, the poor word-spacing of letter-spaced text, and the absence of bold face text. As mentioned above, the TLL comes with its own (Unicode) font, which is used to display the HTML views of the TLL entries. Clearly, this strategy has advantages, as it ensures that the user is able to display the macrons and breves, and the polytonic Greek, Hebrew, and other non-Latin characters that the lexicon contains in abundance. On the other hand, it also has its drawbacks. Firstly, this is not a particularly well-hinted font, so it looks jagged on the screen when compared with the standard Microsoft fonts, which are laboriously and expensively hinted to display very smoothly on screen.

Secondly, the TLL font does not come with italic or bold variants, so this sort of formatting, with which the lexicon abounds, is displayed wrongly or not at all. The problem is less acute than it might be, since Internet Explorer substitutes an oblique roman typeface to compensate for the lack of italic. Thus the information conveyed by the use of italic is not lost, but aesthetically it is not the same thing at all (compare the shape of an italic "a" with a roman "a" that has been artificially slanted). Many readers might not be able to identify this problem precisely, but would nevertheless perceive the resulting lack of polish and professionalism in the output. The absence of bold text is likewise not fatal, since outline view can help to point out the structure of the entry even more clearly than bold face but, since the HTML markup indicates the text that should be bold, it would be nice if the browser would display it that way.

The user does have a way of "fixing" these problems; namely, to un-install the TLL font and force the browser to fall back to using Times New Roman. In this way, you get to see the TLL with true italics and with bold, but you will not see any special, non-standard characters that the TLL font uses to convey information. The most important of these is the crucial asterisk sign (which looks a bit like two baguettes crossed in an "X"), with which the TLL indicates an

article that does not give a comprehensive report of all of the citations of a word found in the texts covered by the lexicon. This symbol is represented by a special character unique to the TLL font, and bizarrely it has been substituted for one of the Tibetan Unicode characters (U+0F3E). This is bad practice, and at the very least, Saur should have used one of the Unicode "private use areas". Even better, Saur could have substituted in the TLL font their special asterisk for a vaguely similar-looking Unicode symbol (e.g. U+2724 or U+2A2F), so that those of use who prefer not to use the TLL font would see something approximating an asterisk, rather than a tiny Tibetan squiggle.

One could argue that providing a special-purpose font is not necessary for most users; indeed, one of the purposes of the Unicode standard was to make such special-purpose fonts obsolete. Most Latinists in this day and age are very likely to have a computer configured with a Unicode font able to display polytonic Greek, Hebrew, Latin vowels with macrons and breves, and the more usual ancient scripts; even if the user has not installed fonts especially for this purpose, modern operating systems now come with very comprehensive Unicode fonts as standard. Saur should try to make the TLL work better with the Unicode fonts the user already has installed, rather than insisting on installing a special-purpose one. It should still, however, provide the TLL font for those users with older computers that may not have such fonts already installed.

Another aesthetic problem with the HTML output is the way letter-spaced text is displayed. Definitions in the TLL commonly use for emphasis lower-case text with larger than normal inter-word spacing, a practice once widespread in Germany, though now universally decried as an abomination against legibility. It is rarely found in modern texts, so it is not surprising that Internet Explorer does a poor job of displaying it, even though it is allowed for in Cascading Style Sheets (CSS), which are used to specify the way the HTML pages are displayed by the browser. The problem, quite apart from the inherently poor legibility of letter-spaced lower-case text, is that the intra-word spacing is so close to the inter-word spacing that it becomes very hard to know where one word ends and the next begins. Likewise, punctuation inside letter-spaced text is also spaced out, so commas and such float between two words. The obvious solution to this problem is for the browser automatically to increase the inter-word spacing in and around these words by a similar proportion, and to treat punctuation specially. Internet Explorer doesn't do this at all, the Gecko engine (used by Mozilla) does it somewhat, and the KHTML engine (used by Safari and Konqueror) does it quite well. Unfortunately, at present, only Internet Explorer can be used with the TLL, unless you are willing to go through a fair bit of manual fiddling each time you look at an entry.

The limitations of Internet Explorer (IE) are present in other ways, too. When there is a superscript in a citation, such as the number of an edition, this throws off the inter-line spacing, and the line number, if any, is displayed next to a blank line. Other browsers handle this just fine. In general, when you open one of the TLL HTML views in another browser, most of the text is displayed correctly, indeed better than in IE. On the other hand, the column number and line number, which are displayed in a column on the left side of the screen, are both run together as one number, which is quite annoying. In IE, the numbers are separated by a space, as they should be, but this only happens because of a bug in that browser. It is wrong to rely on non-standard behavior, and doubly so to rely on a bug in a particular version of a particular browser, as it is entirely possible that later versions of IE will fix this behavior. As it happens, the fix for this problem is trivial.⁸ Another such easily fixed peculiarity is that using other browsers with different XSLT processors will result in stray spaces pervading the output, especially before punctuation. The XSLT processor included with IE removes stray whitespace by default, but not all processors do so, so this should be specified explicitly.⁹

Despite these easy-to-fix flaws, it should be emphasized that, by and large, the HTML output of the TLL is not tied to the peculiarities of Internet Explorer, and displays correctly on any modern browser, though there may be a few quirks in some of the advanced features. For example, the little boxes in outline view that were mentioned above, which tell you whether the given level of the outline is expanded (with a minus sign) or collapsed (with a plus sign) and which toggle their state when clicked, are implemented with JavaScript. In addition to Internet Explorer, I tested them on some non-Windows browsers; they work perfectly on Mozilla Firefox, but have some problems on Konqueror. The conclusion is that there is no major technical obstacle to making a version of the TLL which displays its HTML output on any computer that connects to it via a modern browser; there is no good reason that the TLL should be tied to users of the Internet Explorer browser or indeed of the Windows operating system.

Storage and Encryption

In the foregoing account of the XML source files in which the TLL entries have been encoded and the HTML views of those files which are shown to the user, I omitted to mention one intermediate step. When the user requests an article, the required XML file is not retrieved directly from the disk, but rather from a zip file within which that XML file is stored. Text files such as these XML files compress very well, so this is a reasonable step to take, in order to save disk space on the user's computer. The XML files making up the TLL, which number over 70,000, are stored in compressed form in a set of 13 zip files, which most users will have installed on the hard disk. These are not ordinary zip files, though; they are password protected, so that the user cannot view the contents of any of the XML files they contain.

This is a peculiar step to take. If this encryption was added in an attempt to prevent unauthorized copying of the software by those who have not paid for it, it is an ineffective measure. There is nothing to stop a person from taking the installation media and copying the data by the more straight-forward method of going around to another computer and installing it in the normal way. If the password-protection was added in an attempt to keep the data secret from commercial competitors, the measure is equally weak. The variety of encryption used in zip files was demonstrated to be fundamentally insecure in a well-known 1994 paper by Biham and Kocher. Software that implements the techniques of that paper to reveal the passwords of zip files is easily available on the Internet.^{[10](#)}

If encrypting the zip files does not prevent unauthorized pirating of the software by users and does nothing more than slow down for a few hours a competitor who wants to view the XML source files, then why bother? I do not have an answer for that question, but I can point to the costs of the decision to do so. It means that a project which consists almost entirely of text files, which have been encoded in an open and cross-platform manner, all of a sudden requires specialized client software if those files are to be accessed. Any modern web browser can take an XML file and an XSLT stylesheet and combine them into a web page for the user to view; they cannot do this, however, if that XML file is locked up in a password-protected file. Thus, Saur has to provide the user with a special-purpose program to access the TLL; this program turns out to be vast, bug-ridden and monstrously complex; on account of its complexity, it becomes impossible to port it to another platform; thus the TLL is available only for Windows. Fortunately, as we will see, this overly complex, Windows-only user interface in fact adds little of value, and so it could easily be jettisoned by most users.^{[11](#)}

User Interface

When you start up the TLL program on your Windows computer, you do not see a web browser, as the focus of the comments above might imply; instead you see a bewildering

collection of tick-boxes, text entry fields and result output areas, which looks as if a demented programmer had tried to see how many widgets he could fit on one screen. I would not be surprised if technophobic Latinists simply shut the program down on first sight and gave up on the electronic TLL as unintelligible and unusable. If you persevere, the complexity of the user interface is mitigated somewhat by the thorough on-line help, which explains the use of just about every element on every screen. When you inspect the opening page more closely, you will see that there is a series of "tabs" identifying screens you can choose from:

ToC | Lemmas | Full Text | Keyword | Expert Search | User Preferences | Full Display

The one you are currently looking at is the "Full Text" search page, which is the last thing most users will want. Of all of these tabs, the first is the only one the vast majority of users will ever need to use.

So we start by choosing the "ToC" or Table of Contents tab, and there, after a long wait, we get a scroll box listing the various volumes of the TLL. To find a particular lemma, click on the desired volume to reveal its parts, and so on, until you find the lemma you want. If this has sub-lemmata, you can click on it to reveal them, too. Clicking on a lemma or sub-lemma means that the article comes up, again after a longish wait, in a preview window on the right (provided that the "Show Preview" box is ticked). This window is too small to view the entry comfortably, so click on "Open Article" to open the article in Internet Explorer.

When you view the entry in IE, you see the "article view" in a frame on the right, and on the left is a frame that gives general information about the entry and has links you can click on to change the display on the right to "outline" or "citation" view. There are also arrows on the left which you can click on in case you have arrived here as a result of a search (see below); these will take you to the next hit or previous hit within the entry. The border between the left and right frame can be moved with the mouse to make more room for the text of the entry.

If you want to look at another word, go back to the TLL program, and, if this next word comes from a different volume or part-volume of the TLL, the first thing you should do is to go back to the very top of the list of lemmata and return it to its original state. You do this by clicking on the title of the expanded volume or part-volume to toggle it from open to closed, thus hiding all the lemmata. Then you see only the list of volumes or part-volumes, and you can repeat the process of opening up another volume and part for the entry you require. The reason for proceeding in this way is that, if you consecutively open up lists of lemmata from various volumes one after the other, the ever increasing number of entries in the little scroll-box makes it more and more unwieldy to scroll through to find the lemma you want. This problem is demonstrated most clearly by the neighboring tab, marked "Lemmas". This provides a list, in a scroll-box even smaller than on the "ToC" screen, of all of the lemmata and sub-lemmata in the electronic TLL. I have no idea how many these are, but I would guess about 100,000 or so, all crammed into a two-inch high box; it is a classic example of bad interface design. Trying to find the lemma you want in that little box is nearly impossible.

Another problem with the "ToC" and "Lemmas" tabs is that, when you first access them, it takes an eternity for the screen to come up. I was testing the TLL on an old machine, so I do not object when some computationally intensive process, such as calling up an entry, takes a while to complete: the XML file has to be uncompressed, decrypted, loaded into the browser and parsed; stylesheets and other auxiliary files have to be loaded and parsed; and finally the HTML has to be generated, rendered and displayed. It is not really very surprising that, when calling up a very long entry like "et", all of this takes around 45 seconds on the slow computer I

was using. On the other hand, when it takes over a minute to load the "ToC" or "Lemmas" screen, which does nothing but display a list of words that always stays the same, it cannot be the result of anything other than bad software design. If the time is being spent in generating the list of lemmata and sub-lemmata, then that should only be done once, before the user receives the software. If it is a result of cramming too many choices into a scroll-box that was only designed to hold a modest number, then it highlights what a poor choice of interface this was.

Before leaving the "ToC" and "Lemmas" tab, some of the good things about them should be stressed. The inclusion of sub-lemmata in the lists is extremely useful. In longer entries in the printed edition these are sometimes indicated in the running heads of the pages, but not always, and so figuring out where to find a sub-entry can be difficult. In the electronic TLL, you can jump right to it. Saur have also done a very thorough job with including the many cross-references in the TLL into the list of lemmata. So, for example, when the printed text gives a series of cross-referenced words, all of which point to the same destination, each of these words is separately listed in the list of lemmata.

There are two things that the potential user of the TLL will most likely want to do: look up a particular word, or check a particular citation. The "ToC" tab lets you do the former, and for the latter there is a set of four input fields at the top of the "Full Text" tab where you can put in numbers for volume, part, column and line, if you already have a TLL reference and want to jump right to it. This is very handy in those cases, and the only trick is that, in volumes not subdivided into parts, you must enter a zero for the part number, as the help files explain.

Of the other tabs, "User Preferences" is self-explanatory, and "Full Display" is where the text of the desired article is displayed (unless you use the "ToC"), from which you can open it up in IE. This leaves three tabs in the middle: "Full Text", "Keyword" and "Expert Search". All of these allow you to search for text anywhere in the body of the TLL. All are very complicated screens, and it is clear that a vast amount of effort has gone into providing this functionality. These screens allow a very wide variety of ways to specify a search across the whole of the TLL; then the entries in which the text has been found are listed, and it is possible to open each in turn; then you can jump to the location of each hit in that article. Unfortunately, the program kept crashing when I was testing the search functionality, which I do not think is acceptable, and so I did not test these screens except briefly.

More importantly, I cannot think of any urgent reason why anyone would want to do searches of this sort. If you want to search for text in Latin literary or epigraphical corpora, there are other databases in which it is easy to do that. The whole point of the TLL is that it contains all of classical Latin literature, in a pre-digested form, alphabetically arranged. Within a given article, it is extremely useful to be able to do a search, which you can now do within your web browser, but why would one want to search the body of the entire lexicon? I suppose that there must be some people to whom this functionality is useful, but I doubt there are many. Luehken, in his fine review of the first electronic edition, valiantly attempts to excite us about the potential for this search feature, but the three usage scenarios he suggests all strike me as wholly artificial and unconvincing.¹²

The issue is not that it is bad to have this additional functionality provided to those few who may wish to have it; it is that it seems to have become the tail that wags the dog. The data included with the TLL include a gigabyte of what appear to be index files, whose purpose would seem to be speeding up searches of the entire lexicon. Indeed, global searches are extremely fast, which suggests that the lexicon has been indexed. The problem is the contrast with looking up a particular lemma, which is a much simpler and much commoner task but is

very slow. It is hard to avoid the impression that the designers of the interface software thought that the normal usage would be to search the lexicon electronically, so this needed to be made fast, and that looking up a particular lemma would be an unusual requirement, and so it would not matter if this were slow. In fact, the opposite is the case.

Another problem with this "extra" search functionality is that it has complicated the user interface software to the point where porting it to another platform, other than the Windows-only .NET environment for which it has been written, would be prohibitively difficult and expensive. I am quite convinced that, if Saur were prepared to market the TLL with a radically simpler interface, it could easily be made available in a form whereby it could be installed once, on a university server, and any student or teacher with a modern web browser, regardless of operating system, could access it. Such a user interface would provide only two things: a way to navigate quickly to a given lemma or sub-lemma, and a way to jump to a particular citation of volume, part, column and line. I believe this would satisfy the needs of 99% of users. Anyone who really needed the full-text search functionality could go ahead and install the present user interface on a Windows machine.

A Proposal

At the start of this review, the deficiencies of the current implementation of the TLL were summarized as follows: Microsoft Windows only, slow and cumbersome user interface, missing entries, tedious installation of the networked version. All of these problems could be solved if Saur were to market a site-licensed version of the TLL that could be installed on a web server, which would require of the end-user only that he or she have a modern web browser installed. This would make university computer administrators happy since it would mean installing only one piece of software on one server, rather than individually for users all over campus. It would make end users happy since they could access the TLL from any sort of computer connected to the campus network -- office machine, home machine, laptop, department computer room, classroom, without installing any special software. The only drawback I can see is that this web-based TLL would not include (or at least not at first) the complex full-text search functionality of the current software. The few users who really need this could, however, just install the current Windows-only software, and use that instead.

Finally, a cross-platform, web-enabled TLL should make Saur happy, too. I strongly suspect that the publisher has no idea how strongly entrenched the Macintosh is in North American humanities departments. To judge from my own experience, it seems likely that the majority of professional Classicists in North America use Macs, and many work in departments where there is nary a Windows machine to be found. That is a subjective impression, but it cannot be denied that Mac users are at the very least a very substantial minority of that population. Why would Saur market a product that cannot be used (or can be used only with great difficulty) by such a large and well-heeled part of its target market? In continental Europe, the Mac is much more of a niche item, and so I suspect Saur simply has not realized how many more licenses for the TLL they could sell by appealing to the non-Windows market. There is no reason why Saur should believe me about this, so, if you are a non-Windows-using Classicist working in an institution that would buy a site-licensed, platform-independent, web-based version of the TLL, such as the one described here, send them an e-mail and let them know that they are missing out on a sales opportunity (saur.info@thomson.com).

The other side of the equation for the publisher is cost. Here, the excellent technical foundations of the TLL would pay off handsomely. Since Saur has relied on open standards to encode and display the entries, it is already the case that a Mac or Linux user can view the output of the TLL via a browser such as Mozilla Firefox. All that is needed is a small bit of

work to develop a web-based interface that would allow the user to do two things: to select a lemma or sub-lemma from a list and to enter a citation by volume, part, column and line number. The rest of the work in displaying the TLL as web pages on any modern computer has in effect already been done by Saur. To develop a prototype of this web application should not take an experienced programmer much time at all, as the techniques are well understood.¹³

To conclude, I want to stress the exemplary job Saur has done in digitizing the TLL. The important technical decisions were made wisely, and the vast task of overseeing the transcription and encoding has been done very accurately. By using durable, open standards to encode and display the TLL, Saur has ensured a long future for this venerable and essential publication. The entire project is very close to being perfect, except for some flaws in the current user interface. The availability of the TLL in electronic form is a major event in Latin studies, and it has already enabled students of the language to use this tool in valuable new ways. Let us hope that in the near future Saur brings out a version that will be easier for everyone to consult.¹⁴

Appendix

This problem of mis-sorting the numerical citations was noted by Luehken (op. cit. n. 6, p. 1114, n. 33). The problem is that the comma-separated numbers get sorted alphabetically rather than numerically. The problem can be fixed for the most common cases by means of a crude workaround (which would be slightly cleaner if one were to use the new "tokenize" command from version 2.0 of the XSLT specification). You should only install this workaround if you know what you are doing. At both line 197 and 213 of the file "ttscitation.xsl", replace the line that reads

```
<xsl:sort select="normalize-space(location)"/>
```

with the following lines

```
<xsl:sort select="number(location)" data-type="number"/>
<xsl:sort select="number(substring-before(location, ','))" data-type="number"/>
<xsl:sort select="number(substring-after(location, ','))" data-type="number"/>
<xsl:sort select="number(substring-before(substring-after(location, ','), ','))" data-
type="number"/>
<xsl:sort select="number(substring-after(substring-after(location, ','), ','))" data-
type="number"/>
<xsl:sort select="normalize-space(location)"/>
```

This only accounts for proper sorting of regular citations of the form "1", "1, 2" and "1, 2, 3"; any citations that deviate from that form (and very many do) will still be wrongly sorted.

Notes:

1. The canonical explanation of the principles of the TLL is the "Praemonenda de rationibus et usu operis", which was published in many languages, including English, as part of the TLL in 1990; it is planned that this will be added to the electronic TLL in the next release.

2. http://www.saur.de/_download/prospekte/englisch/0000011514.pdf

3. <http://www.maiierphil.de/>

4. See <http://www.tei-c.org/Applications/el02.xml>. The markup is clearly not completely TEI-

conforming, but much of it is intelligible to someone familiar with the TEI. The markup may be deduced from the DTD, which is helpfully included with the software (in the file called "thesaurus32.dtd"), and which has some useful comments. For example the comment on the "lemma" element reads "Body of the lemma. Contains: 'def' (definition of the headword), 'cit' (citation), 'sense' (to represent the hierarchical view of the body; the attribute 'level' indicates the level for each sense: ex. 1.1 or 1.2.1)." There is a minor error in the DTD: the "role" attribute of the "cell" element is defined twice. To clarify the comment in the DTD regarding the <sense> element, the distinctive, deeply nested structure of TLL entries is encoded as e.g. <sense level="1.12.2.2.2.22">, where the position of the current definition in each level of the hierarchy is given by the numbers; this is in line with the suggestions in section 12.2.1 of the P4 TEI guidelines. The bulk of each entry is made up of citations, which are marked up with a <cit> tag, which contains an <xr> tag, giving the cross-reference; bare citations without quoted text are marked-up with the <xr> tag alone. These are keyed (using the "target" attribute of the <xr> element) to a numbered scheme of authors and works such that clicking on the link in the HTML version of the entry takes you to a bibliographical account of the text in the Index Librorum.

5. E.g. <http://www.perseus.tufts.edu/cgi-bin/resolveform?lang=Latin>

6. This is the concluding suggestion of the very useful review by Henning Luehken of the first release of the electronic TLL, in *Goettinger Forum fuer Altertumswissenschaft* 6 (2003) 1103-1121, which is available on-line (<http://www.gfa.d-r.de/dr,gfa,006,2003,r,13.pdf>). Before such an intricately hyperlinked Latin library can come into being, we need a standardized way of referring to ancient texts; this is a problem that the Classical Text Services Protocol attempts to address (<http://chs75.harvard.edu/projects/diginc/techpub/cts>). The markup of the TLL would need to be adjusted to take advantage of such a scheme, so it is not a hope for the near term, but is perhaps something to aim for in the future. Luehken also notes the problems with printing TLL articles from a browser (pp. 1118f).

7. Luehken (op. cit. n. 6), p. 1114, n.32.

8. The two numbers are in adjacent elements, without whitespace between them, with style attributes that specify a fixed width for each. IE presumably uses that width to separate the numbers, each in its own fixed-width box, but the CSS specification (version 2.1, section 10.3.1) stipulates that the width property does not apply to inline, non-replaced elements like . So when other browsers run the numbers together, they are implementing the correct behavior. To fix this, in line 162 of the file "ttsarticle.xml", put a just before the first . Another problem with the line numbers is that, in article view, the appearance of a new column is not obvious, as it is on the printed page, so the line and column number should be printed explicitly for the first line of every column. Thus, in line 162 of "ttsarticle.xml", "\$row mod 5 = 0" should be corrected to read "\$row mod 5 = 0 or \$row = 1".

9. The following line should be added to the file "ttscommon.xml": <xsl:strip-space elements="*" />

10. Biham, E. and Kocher, P. "A Known Plaintext Attack on the PKZIP Stream Cipher." In *Fast Software Encryption: Second International Workshop*, Leuven, Belgium, 14-16 December 1994, Proceedings. Springer: Lecture Notes in Computer Science 1008. A PostScript version of the paper is available on-line at: <http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/1994/CS/CS0842.PS> The technique described in that paper requires a small decrypted sample of the encrypted text. In this case, that would be a decrypted XML file, easily obtained from a memory dump of a computer running the TLL software.

11. Luehken (op. cit. n. 6, pp. 1107, n. 11) comments on the pointed contrast between the open standards employed in the encoding of the electronic TLL, and the closed platform on which the user interface is based.

12. Luehken (op. cit. n. 6), pp. 1115-7.

13. The interface could consist of an opening page that allows the user to enter a precise location in the TLL or, if looking for a lemma, to select a particular volume. In the latter case, the user would navigate a series of web pages until the lemma was found, somewhat like the

way the "ToC" tab of the current interface works. Then the required article would be displayed just as it is now. Most of this could be implemented as a simple collection of static HTML, XML and XSLT pages, but it would be necessary to generate some content dynamically. The internal cross-references from one part of the TLL to another, which would also enable the user to enter a precise TLL location, and the citation labels, which take you to the corresponding entry in the Index Librorum, all need to be translated from the scheme used to encode them to another scheme that would give the location of the cross-referenced web page. This might be done on the client side, using JavaScript; the present implementation seems to use a combination of JavaScript and Active-X, of which the latter is Windows-only and would have to be replaced. On the other hand, this translation of cross-references could also be done on the server; the most obvious technique would be to use a CGI script, which would be very easy to install on a wide variety of web servers. If it is the case that Saur has a bureaucratic requirement to keep the XML files bundled up in password-protected zip files, despite the demonstrable futility of this measure, then that requirement can be accommodated by writing the web interface as a CGI script in a portable, compiled language, which would hide away the password, and would unzip each of the XML files on demand. There may, of course, be some complications I have not foreseen, since I have not been able to read the source code of the user interface. Some things I am not sure of are: how the list of lemmata and sub-lemmata is drawn up, and how the internal links to citations by volume, part, column, and line are mapped to entries.

14. Many thanks are due to Kathy Coleman and Harry Hine for their assistance with this review, and to Irmgard Schaefer at K. G. Saur for answering my questions most helpfully.

[Read
Latest](#)

[Index for
2006](#)

[Change Greek
Display](#)

[Archives](#)

[Books Available for
Review](#)

[BMCR
Home](#)

HTML generated at 13:30:45, Friday, 03 April 2009